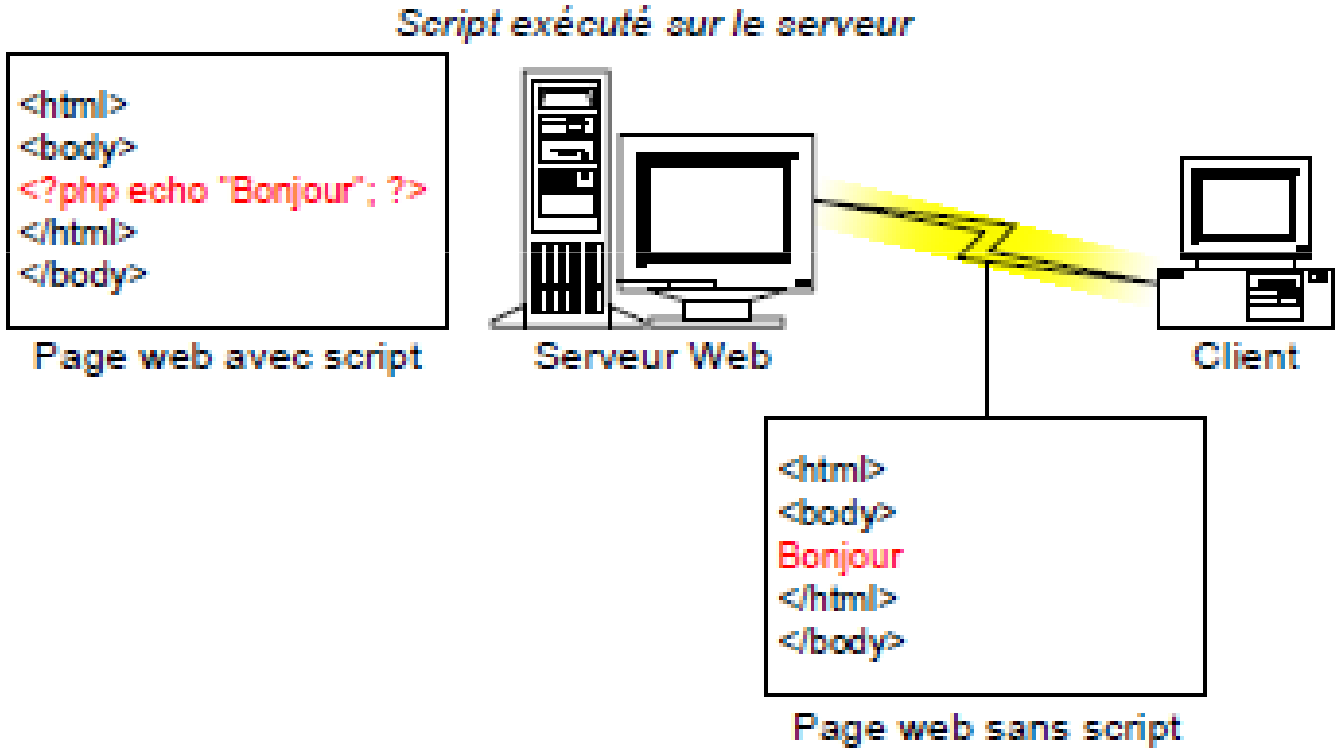
Le langage PHP

M.BOUABID, 03-2014

Généralités (1/2)

- **PHP** Hypertext Preprocessor
- créé en 1994 par Rasmus Lerdorf. A l'époque, PHP signifiait *Personnal Home Page*
- C'est un langage de programmation web, exécuté du côté serveur , qui permet la création des pages web dynamiques
- Version actuelle : PHP 5.5.10 (6 mars 2014)
- Syntaxe du langage PHP est proche de celle de C,Perl ou Java
- Gratuit, fonctionne sous Unix et windows

Généralités (2/2)



Syntaxe de base

- Un code PHP doit être placé entre les balises php

```
<?PHP
print 'bonjour';
?>
```

- Séparateur d'instruction ;
- Commentaries
  - //toute une ligne
  - /*plusieurs
    ligne*/

Variables et types (1/3)

- Variables : préfixées par le caractère $
- Php ne necessite pas de déclaration explicite du type de variable
- Types de données__Ex.d'affectation
  - Nombres entier_____$i=1;
  - Nombres réelles_____$pi=3.14;
  - Chaine de caracatère____$ch="oui"
  - Conversion de type : « cast » comme en C
    - $x=2.0, $y=(int)$x;// y=2

Variables et types (2/3)

- Exemple 1 : <?php

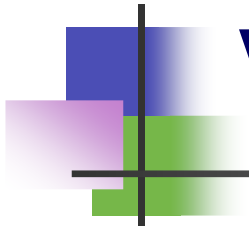
```
$nom = 'visiteur';
echo"bonjour $nom";  ?>
```

- Exemple 2 :<?php

```
$nom = 'visiteur';
echo'bonjour '.$nom; ?>
```

- Exemple 3 : <?php

```
$nom = 'visiteur';
echo'bonjour $nom';  ?>
```



Variables et types (3/3)

- Quelques fonctions :
  - **empty($var)** : renvoie vrai si la variable est vide
  - **isset($var)** : renvoie vrai si la variable existe
  - **unset($var)** : détruit une variable

Les constantes –les variables d'environnement (1/2)

- Les variables d'environnement
  - Pour connaître les différentes variables d'environnements, il suffit de faire une page php contenant ceci

```
<?php  phpinfo(); ?>
```

  - Exemples :
    - HTTP_HOST //nom de domaine du serveur : phpdebutant.org
    - $_SERVER['SERVER_ADDR'] //adresse IP du serveur

Les constantes –les variables d’environnement (2/2)

- $DOCUMENT_ROOT // le répertoire racine de l'arborescence des document sur le serveur
- $_SERVER['HTTP_USER_AGENT']// type de navigateur
- $REQUEST_METHOD //la méthode utilisée , GET ,POST , pratique pour vérifier les variable provenant d'un formulaire

Les constantes –les constantes définies par l'utilisateur

- Les constantes définies par l'utilisateur
  - Exemple :
    - define ("unechaine", "valeur_de_de_chaine"
    - define ("pi", "3.14159265")
    - echo " unechaine =" . unechaine. "
";
    - echo " pi=".pi. " "

Les opérateurs (1/2)

- **Opérateurs arithmétiques :**

+ (addition), **-** (soustraction), ***** (multiplié), **/** (divisé), **%** (modulo), **++** (incrément), **--**(décrément). Ces deux derniers peuvent être pré ou post fixés

- **Opérateurs d'assignement :**

= (affectation), ***=** (**$x*=$y** équivalent à **$x=$x*$y**), **/=**, **+=**, **-=**, **%=**

- **Opérateurs logiques :**

and, **&&** (et), **or**, **||** (ou), **xor** (ou exclusif), **!** (non)

- **Opérateurs de comparaison :**

== (égalité), **<** (inférieur strict), **<=** (inférieur large), **>**, **>=**, **!=** (différence)

Les opérateurs (2/2)

- Concaténation des chaines de caractères
  - Exemple
    - $ch1=$ch2.$ch3;
    - $ch1.="
"
- Caractères spéciaux dans les chaines
  - Antisalsh : \\
  - Dollar : \$
  - Guillemets: \"

Les conditions (1/2)

- exemple 1, avec if else elseif

```
$variable = 'voiture';

if($variable = = 'voiture'){
print 'bravo vous avez trouvé';
}
elseif($variable = ='automobile'){
print 'c\'est presque ca';
}
else {
print 'ce n\'est pas ca veuillez rééssayer';
}
```

Les conditions (2/2)

- exemple 2 switch()

```
switch($operation)
{
case '1': // si la variable opération est égale à 1
print ' operation numero 1'; // on affiche cette phrase
break; // on referme cette condition

case '2':// si la variable opération est égale à 2
print 'operation numero 2';
break;

default: // si la variable opération n'est pas égale à 1 ni à 2 ou si
elle n'est pas définie
print 'operation par defaut'; // on affiche une phrase par défaut
}
```

Les boucles (1/2)

- exemple avec while ()

```
$i= 0; // on défini une variable à 0 pour le compteur de boucle

while ( $i < '4' ) // la boucle s'arretera lorsque la variable $i sera égale à 4
{
print 'boucle numero '.$i.'<br>'; // on affiche une phrase avec le numero de la boucle

$i++; // le ++ sert à ajouter 1 à chaque tour de boucle, ne l'oubliez pas sinon la boucle sera infini donc affichera une erreur !

}
```

Les boucles (2/2)

- exemple avec for()

```
for ($i=0;$i<4;$i++)
{
print 'boucle numero '.$i.'<br>'; // on affiche une phrase avec le
numero de la boucle
}
```

Affichera à l'écran

boucle numero 1
boucle numero 2
boucle numero 3
boucle numero 4

Les tableaux (1/2)

- Il existe 2 types de tableaux, les tableau nominatifs et les associatifs.
- Exemple de tableau nominatif:

```
<?
$tableau = array('az-php','php4','mysql'); // on
déclare les valeurs du tableau
print $tableau;
?>
```

Les tableaux (2/2)

- exemple de tableau associatif:

```
<?
$tableau_ass = array(site=>'az-
php',language=>'php4',base=>'mysql');
// on déclare le tableau sous le format
nom_de_variable=>'valeur'

//pour afficher le site par exemple
print 'le nom du site est '.$tableau_ass[site];
?>
```

Se servir des boucles pour parcourir un tableau Avec foreach() (1/2)

```
$tableau = array('az-php','php4','mysql'); //on défini le tableau et ses éléments

foreach ( $tableau as $contenu ) //on parcours le tableau
{
print $contenu.'<br>';//on affiche le contenu
}
```

Affiche à l'écran
az-php
php4
mysql

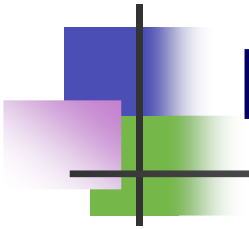
Se servir des boucles pour parcourir un tableau Avec foreach() (2/2)

```
tableau_ass = array(site=>'az-
   php',language=>'php4',base=>'mysql');

   foreach ( $tableau as $cle=>$val ) //on parcours le tableau
   {
   print $val.'<br>';//on affiche le contenu
   }
```

Les tableaux -quelques fonctions

- **count($tab)**, **sizeof** : retournent le nombre d'éléments du tableau
- **in_array($var,$tab)** : dit si la valeur de **$var** existe dans le tableau **$tab**
- **list($var1,$var2…)** : transforme une liste de variables en tableau
- **sort($tab)** : trie alphanumérique les éléments du tableau



Les tableaux -quelques fonctions

- **implode($str,$tab), join** : retournent une chaîne de caractères contenant les éléments du tableau **$tab** joints par la chaîne de jointure **$str**

- **explode($delim,$str)** : retourne un tableau dont les éléments résultent du hachage de la chaîne **$str** par le délimiteur **$delim**

Inclusion des fichiers dans une page

- Fonction require() : provoque une erreur fatale si le fichier requis manque (interruption de script)
- Fonction include : provoque seulement un avertissement (warnind) si le fichier requis manque
- Exemples (paramètres des fonctions : un URL
  - Require ("mes_fonctions.php");
  - Include("unepage.html");

Les fonctions (1/2)

- Squelette de définition
  - Function nom_fn($p1,$p2,…,$pn){

```
        //code de la fonction
    return val_retour;//optionnel
}
```

  - Exemple

```
Function max($n1,$n2){
Return ($n1>$n2)?$n1:$n2;
}
```

Les fonctions (1/2)

- Passage des paramètre par références :&
  - Fonction alaligne(&$chaine){

```
$chaine.="<br >"
}
$ch="coucou" ;
Alaligne($ch);
```

Les fonctions (2/2)

- Passage par valeur (passage par défaut)
  - Function cafe ($type="express"){

```
 return  je fais un $type<br/>« ;
}
Echo cafe();//je fais un express
Echo cafe(" capucino");//je fais un capucino
```

Visibilités des variables

- Variables normales : visible uniquement à l'intérieur de la fonction où elle sont définies
- Variables globales
  - Function f(){

    global $vglobal; //visible hors de la fonctions

    }
- Variables statiques
  - function g(){ static $vstat;}//conserve sa valeur entre deux appels à la fonctions

Séparation de PHP et HTML

- Séparation des tâches de programmation et de design web
- Maintenances facilitée
  - Lecture plus aisé des pages HTML+PHP
  - Modification facile du code sans affecter la mise en page et vice-versa
- Réutilisation de code
  - Utilisations des fonctions include() et require()
    - Ex. définition dans des fichiers séparés d'entêtes et de pied de page communs à plusieurs pages